

MVAPICH2 2.3.7 Quick Start Guide

MVAPICH TEAM

NETWORK-BASED COMPUTING LABORATORY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
THE OHIO STATE UNIVERSITY

<http://mvapich.cse.ohio-state.edu>

Copyright (c) 2001-2022
Network-Based Computing Laboratory,
headed by Dr. D. K. Panda.
All rights reserved.

Last revised: March 2, 2022

1 Overview

This Quick Start contains the necessary information for MVAPICH2 users to download, install, and use MVAPICH2 2.3.7. Please refer to our [User Guide](#) for the comprehensive list of all features and instructions about how to use them.

MVAPICH2 (pronounced as “em-vah-pich 2”) is an *open-source* MPI software to exploit the novel features and mechanisms of high-performance networking technologies (InfiniBand, iWARP, RDMA over Converged Enhanced Ethernet (RoCE v1 and v2), Slingshot 10, and Rockport Networks) and deliver best performance and scalability to MPI applications. This software is developed in the [Network-Based Computing Laboratory \(NBCL\)](#), headed by [Prof. Dhabaleswar K. \(DK\) Panda](#) since 2001.

More details on MVAPICH2 software, users list, mailing lists, sample performance numbers on a wide range of platforms and interconnects, a set of OSU benchmarks and related publications can be obtained from our [website](#).

2 Build MVAPICH2 from Source

The MVAPICH2 2.3.7 source code package includes MPICH 3.2.1. All the required files are present in a single tarball.

2.1 Download & Unpack

Download the most recent distribution tarball from <http://mvapich.cse.ohio-state.edu/download/mvapich/mv2/mvapich2-2.3.7.tar.gz>

```
$ wget http://mvapich.cse.ohio-state.edu/download/mvapich/mv2/mvapich2-2.3.7.tar.gz
$ gzip -dc mvapich2-2.3.7.tar.gz | tar -x
$ cd mvapich2-2.3.7
```

2.2 Configure

If you’re using a Mellanox InfiniBand, RoCE, iWARP, Slingshot, or Rockport Networks network adapter you can use the default configuration...

```
$ ./configure
```

If you’re using a Intel TrueScale InfiniBand adapter or Intel Omni-Path adapter you should use...

```
$ ./configure --with-device=ch3:psm
```

2.2.1 Other Configure Options

—**prefix** This option tells the build system where to install mvapich2. If this option is not given mvapich2 will be installed in /usr/local.

—**disable-shared** This option tells the build system to create static libraries only. By default, both the shared and static libraries are built and installed.

—**enable-g=all** —**enable-error-messages=all** This option controls the amount of debugging information available in the MPI library. By default these are disabled since this will affect the size and speed of the MPI library.

2.2.2 More Options

MVAPICH2 supports many other configure and run time options which may be useful for advanced users. Please refer to our [User Guide](#) for more complete details.

2.3 Build & Install

```
$ make -j4          # parallel build
$ make install
```

3 Run MPI Program

In this section we will show how to build and run a hello world program which uses mpi.

3.1 Build & Run

```
$ mpicc -o mpihello mpihello.c <1>
$ mpirun_rsh -hostfile hosts -n 2 ./mpihello <2>
```

1. `mpicc` is one of the basic commands used to compile MPI applications. This along with `mpiCC`, `mpif77`, and `mpif90` are wrapper scripts that invoke the compiler used to compile the MVAPICH2 library. Use of these scripts are recommended over invoking the compiler directly and adding the `CFLAGS` and `LDFLAGS`
2. `mpirun_rsh` is used to launch MPI programs. This command tells `mpirun_rsh` to launch 2 `./mpihello` processes using the nodes specified in the hostfile `hosts`.

3.1.1 Using `mpirun_rsh`

syntax

```
mpirun_rsh <options> <env variables> <command>
```

options

-hostfile specify the location of the hostfile

Hostfile Format The `mpirun_rsh` hostfile format allows for users to specify hostnames, one per line, optionally with a multiplier, and HCA specification. The multiplier allows you to save typing by allowing you to specify blocked distribution of MPI ranks using one line per hostname. The HCA specification allows you to force an MPI rank to use a particular HCA.

The optional components are delimited by a `‘:’`. Comments and empty lines are also allowed. Comments start with `‘#’` and continue to the next newline.

The following demonstrates the distribution of MPI ranks when using different hostfiles:

Examples:

```
hosts1 node1
        node2
```

```
hosts2 node1
        node1
        node2
        node2
```

```
hosts3 node1:2
        node2:2
```

```
Output of mpihello with different hostfiles $ mpirun_rsh -hostfile hosts1 -n 4 ./mpihello
rank 0 on node1 says hello!
rank 1 on node2 says hello!
rank 2 on node1 says hello!
rank 3 on node2 says hello!
```

```
$ mpirun_rsh -hostfile hosts2 -n 4 ./mpihello
rank 0 on node1 says hello!
rank 1 on node1 says hello!
rank 2 on node2 says hello!
rank 3 on node2 says hello!
```

```
$ mpirun_rsh -hostfile hosts3 -n 4 ./mpihello
rank 0 on node1 says hello!
rank 1 on node1 says hello!
rank 2 on node2 says hello!
rank 3 on node2 says hello!
```

-n Number of mpi processes to launch. Must be the last option specified

env variables Environment variables are specified using the ‘NAME=VALUE’ syntax directly before the command name is specified.

Pass an environment variable named *FOO* with the value *BAR*

```
$ mpirun_rsh -hostfile hosts -n 2 FOO=BAR ./mpihello
```

3.1.2 Other Launchers

MVAPICH2 supports other launchers and resource managers such as Hydra, SLURM, and PBS. Please look at our [User Guide](#) for more complete details.

4 More Information

Please see the following for more information.

- [User Guides](#)
- [OSU Micro-Benchmarks](#)
- [FAQ](#)